Flexible Corpus Annotation with Property Grammars

Philippe Blache, Marie-Laure Guénot

Flexible Corpus Annotation with Property Grammars

Philippe Blache & Marie-Laure Guénot
LPL, Université de Provence
29 Avenue Robert Schuman
13621 Aix-en-Provence, France
pb@lpl.univ-aix.fr

Abstract

We present in this paper a syntactic annotation project relying on a linguistic formalism called Property Grammars. This constraint-based formalism allows to encode different levels of syntactic information (the granularity description can be tuned according to the set of constraints to be verified). Moreover, this approach allows to propose a description whatever the form of the input (being it grammatical or not). After a presentation of the formalism and its representation, we describe its use in the annotation of a spoken language corpus.

1 Introduction

For most of annotation projects, syntactic information is limited to bracketing. One of the reasons is that enriching corpora with finer syntactic information necessitates by definition the choice of a formalism and a syntactic theory. This problem doesn't necessarily occur with lower levels of annotation such as POS or even bracketing. Moreover, there is another preliminary question when creating a treebank with fine syntactic annotation: in what perspective such a resource is created? Several answers can be given, the most important one consisting in a kind of validation of the theory itself. Annotating a corpus means a confrontation of the theory with unrestricted texts or even (as we will see later) spoken language data. This means the development of large-scale resources which constitutes in the end a systematic descriptive work within a formal framework.

The problem is that usually, the result of this work is mainly interesting and useful for people working in the same theoretical paradigm. Among the set of problems that syntactic annotation has to solve (e.g. encoding ambiguity), this question of reusability is of deep importance. This means in particular that, whatever the formalism or the theory, there always exists some information that is interesting and that can be used in other approaches. For example, the notion of constituent being shared (in one way or another) by generative approaches, bracketing information can then be extracted from treebanks encoded with different theories from this paradigm. But we can also imagine that other kind of information could also be reused, whatever the formalism: this is for example the case of some construction descriptions (e.g. clefts, dislocations, etc.) or even more local relations.

On the other hand, creating a treebank is obviously time-consuming. One important point concerns then the possibility of doing the job, even automatically, in an incremental way. In other

words, we should propose a process allowing to enrich progressively the corpus, such enrichment being for example dependent from the development of a grammar. In this perspective, it is interesting to conceive the annotation task as a several pass process, each pass being possibly independent from the others. Such a process is incremental in the sense that new information can be added to oldest one without redoing the entire process.

To summarize, reusability has to be conceived both from the user side (extracting syntactic information from corpora annotated under different theories) and from the developer one (information has to be added or completed progressively). We present in this paper some elements of answer to this problem. The project described here concerns the development of a French treebank annotated under the formalism of Property Grammars (see [Blache00b]). We propose an annotation formalism encoding information in terms of graphs.

2 Information to be encoded in Property Grammars

We think that a constraint-based approach provides efficient solutions to the question addressed above, in particular because information is represented in a very modular way. This is the case with *Property Grammars* in which syntactic information is represented as a constraint system. In this section, we give an overview of property grammars and, in a second part, their representation by means of graphs, which plays an important role in the annotation process.

2.1 An overview of Property Grammars

In this approach, a constraint is typically a relation between two (or more) categories. We use the set of following constraints: *linearity, dependency, obligation, exclusion, requirement* and *uniqueness*. In a phrase-structure perspective these constraints participate to the description of a phrase. The following figure roughly sketches their respective roles, illustrated with some examples for the *NP*.

Constraint	Definition	Example
Linearity ($\prec$)	Linear precedence constraints.	$Det \prec N$
Dependency ($\rightsquigarrow$)	Dependency relations between categories.	$AP \rightsquigarrow N$
Obligation (*Oblig*)	Set of compulsory and unique categories. One of these categories (and only one) has to be realized in a phrase.	$Oblig(NP) = \{N, Pro\}$
Exclusion ($\not\Rightarrow$)	Restriction of cooccurrence between sets of categories.	$N[pro] \not\Rightarrow Det$
Requirement ($\Rightarrow$)	Mandatory cooccurrence between sets of categories.	$N[com] \Rightarrow Det$
Uniqueness (*Uniq*)	Set of categories which cannot be repeated in a phrase.	$Uniq(NP) = \{Det, N, AP, PP, Pro\}$

This set of properties allows to encode syntactic information. The main characterisitics of such a representation is that all the properties (i.e. all the constraints) are at the same level. At the difference of classical generative approaches in which constituency plays a fundamental role (one have to build first a structure and then verify its properties), properties in *PG* can be verified independently from each other. In this perspective, constituency doesn't play any role and this information is not anymore represented in the grammar. We will see that a category is

described with a set of properties which are relations between different other categories. This last set of categories constitutes then implicitely the set of constituents.

In this approach, its is also possible to verify only a subset of properties without modifying neither the general conception nor the implementation of the system. We can make use of this characteristics in order to select the granularity level of the parse: it is possible to use the same system for shallow parsing or deep parsing in choosing the set of constraints to be verified (cf. [Balfourier02]).

Using constraints in order to describe the properties of an input allows a very flexible way of representing information. In this approach describing an input consists in evaluating the set of constraints for the categories that can be associated to this input. A description is then constituted by the state of the constraint system after such an evaluation which means the set of satisfied and violated constraints. In case of grammatical structures, all constraints are satisfied, otherwise, the description also contains violated constraints. Any kind of input, whatever its form (in other words, being it grammatical or not) can the be parsed in the sense that one can describe its properties. We call in our approach such a description a *characterization*. This notion subsumes grammaticality and allows to replace the question "*is this input grammatical?*" with "*what can we say about this input*?".

Let's take an example from a spoken french corpus. For simplicity, we only focus here on the case of the *NP* which can be (roughly) described by the following subset of properties:

- *Linearity:* *(1) Det $\prec$ N; (2) Det $\prec$ AP; (3) N $\prec$ AP; (4) N $\prec$ PP*
- *Requirement:* *(5) N[com] $\Rightarrow$ Det; (6) NP $\Rightarrow$ Conj*
- *Exclusion:* *(7) N $\not\Rightarrow$ Pro; (8) N[prop] $\not\Rightarrow$ Det*
- *Dependency:* *(9) Det $\rightsquigarrow$ N; (10) AP $\rightsquigarrow$ N; (11) PP $\rightsquigarrow$ N*
- *Obligation:* *(12) Oblig(NP) = $\{$N, Pro, AP, Conj$\}$*

We can notice, in this brief description of the french *NP* properties, the fact that some categories belong to several properties whereas some others only appears in one relation. This is for example the case of embedded *NP* which can only appear, in this description, with a coordination (property [6]). As for this construction, we propose to present the conjunction as an obligatory constituent (i.e. the head of the *NP*) in a coordination. The only relation in which an embeded *NP* appears is the requirement stipulating that a *NP* can occur into another higher *NP* together with a conjunction. This aspect also illustrates the fact that no constuency information is required in the description of a category.

From this set of properties, we can give the characterizations of the different *NP*. As explained before, a characterization is the state of the constraint system after evaluation. It is then formed with satisfied and violated constraints, respectively represented by the sets $\mathcal{P}^+$ and $\mathcal{P}^-$. The constraints in the following examples are indicated by their indexes.

Example :

```
dans la marine tu as droit short blanc chemisette blanche
in the Navy you get white short white shirt
```

Four *NP* participate to the description of this input, one of them being of higher level:

Category	Input	Properties
NP_1	la marine	$\mathcal{P}^+$={1, 5, 7, 9, 12} $\mathcal{P}^- = \emptyset$
NP_2	short blanc	$\mathcal{P}^+$={3, 7, 10, 12} $\mathcal{P}^-$={5}
NP_3	chemisette blanche	$\mathcal{P}^+$={3, 7, 10, 12} $\mathcal{P}^-$={5}
NP_4	NP_2 NP_3	$\mathcal{P}^+$={12} $\mathcal{P}^-$={6}

The first *NP* is positiviley characterized, it satisfies all its relevant properties. NP_2 and NP_3 partly satisfies the set of constraints. In both cases, a requirement property (stipulating that a determiner has to be realized together with the noun) is violated. To its turn, NP_4 doesn't satisfy a requirement property concerning the realization of the conjunction. These last cases illustrates the possibility of describing any kind of inputs, even those that can be considered as ill-formed with respect to the grammar.

In conclusion, property grammar consitutes a flexible and robust approach for representing any kind of syntactic information, at any level and for any kind of input. This represents in itself an important interest in a syntactic annotation perpective.

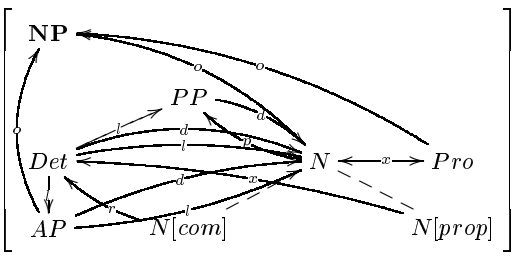
2.2 Representation by means of graphs

We propose in this section a particular representation of syntactic information preserving the flexibility of property grammars. This representation relies on the fact that a property grammar is a set of relations between different objects. A set of properties can be represented as a graph in which categories constitute the nodes and constraints corresponds to the edges.

The graph in [1] represents the set of constraints describing the *NP*. In this representation, the types of the constraints are indicated as graph labels (d for dependency, x for exclusion, l for linearity, o for obligation, r for requirement).

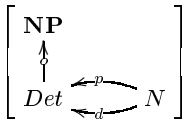
This representation precises the role of the obligation relation which associates a category to its head. This is the only hierarchical relation. The category target of this relation is considered as the root of the graph and corresponds to the phrasal category described by the graph. We indicate this categoy in bold for readability reasons.

(1)



The same representation can be used for the description of a given input. Such a description consists in the set of relevant constraints (i.e. constraints that can be evaluated for a set of categories). The description of a category in context is then formed by the set of constraints that can be evaluated. Such graphs are called *description graphs*. The example [2] presents the

description graph for NP_2 from the example of the previous section.

(2) 

One of the interest of graphs for syntactic annotation (and more generally for the representation of syntactic information) lies in the fact that any information can be represented separately. In this perspective, one can choose to represent only some of the properties. This can be useful for example in shallow parsing.

To summarize, a property grammar is a set of graphs and the characterization of an input is formed by a set of description graphs.

3 Encoding Description Graphs

We have seen in the previous section how graphs can efficiently represent syntactic information from property grammars. We propose in this section to adapt such a representation to the annotation graph approach. This formalism, described in [Bird99], can be used for annotating different kind of information, even heterogeneous. It can for example be used for annotating a corpus with prosodic, phonologic and syntactic information (see for example [Blache00b]). One of the other interests of graphs lies in the possibility of representing partial information, which is interesting in the annotation of spoken language data. Lastly, we have seen in the previous section how non-grammatical inputs can be described by means of description graphs. Annotating a corpus with property grammars consists then in specifying the description graphs characterizing the input. Representing description graphs by means of annotation graphs is straightforward.

3.1 XML representation

There are several solutions for representing such graphs. One consists in representing all information by means of edges. Each property corresponds to an edge, an edge connects in this case different kind of nodes: positions or other categories (which implies the use of labels as nodes). We have choosen here another solution factorizing this information and in which an edge corresponds to a category.

An annotation is an edge connecting different nodes which correspond, at the first stage of the annotation, to positions between words. We will see in the next subsection the interet of such a segmentation and how to generalize it for annotating different level of information. An edge doesn't have in itself a specific semantic other than the localisation of the information in the input (or, in case of audio data, in the signal). In particular, an edge doesn't represent in itself any hierarcical information: the elements dominated by an edge (including other edges) are not necessarily constituents of the object which is described. All the information is contained by a complex structure that can be associated to the edge. We call this structure the *edge description*.

To sumarize, each object is specified with an edge that can bear information. An edge is characterized by the nodes it connects and its (complex) description. At this level of annotation, all edges specify a category. In this perspective, an edge description has to contain different kind

of information, in particular the type of the object it describes, its reference, the set of nodes it connects and, eventually, the characterization of the category. In an XML representaiton, an edge description is an element of the form:

```
<Cat, Index, Feat, Nodes, Charac />
```

in which the arguments are defined as follows:

Cat:	a syntactic category
Index:	an integer referencing the edge
Feat:	a feature vector specifying some morpho-syntactic properties
Nodes:	the set of nodes connected (the constituents)
Charac:	the characterization of the category

In the following, the label of an edge is formed by the category it describes and its index. A label represents then a category that is instanciated or realized in the structure. The feature vector represent some morpho-syntactic information such as agreement, subcategorization, etc. This information is propagated from lexical categories to syntagmatic ones. Another important remark concerns the argument *Nodes* which specifies the set of nodes connected by the edge. This allows the representation of complex information, for example in the case of long-distance or discontinuous relations. This also allows the possibility of representing hyper-edges which can be useful for meta-level information or cross-dependencies between different domains (problem not addressed here). Finally, the *Charac* argument, which contains the characterization, specifies the different properties that are satisfied or violated (i.e. the sets $\mathcal{P}^+$ and $\mathcal{P}^-$). These properties are relations between different categories which are in this approach represented by labels.

The following example illustrates the representation of the different *NP* of the section 2.1. Nodes are indicated with indexes which are positions between words. We indicate by Det_1, N_2, ... the labels representing categories involved in this description. In the same way as for the previous example, and for readbility reasons, properties are also represented by indexes. They should have been indicated with instantiated categories.

```
<NP, 7, <nc-fs->, {Det1, N2}, [P+={1,5,7,9,12} P- = ∅] />
<NP, 8, <nc-ms->, {N3, Adj4}, [P+={3,7,10,12} P-={5}] />
<NP, 9, <nc-fs->, {N5, Adj6}, [P+={3,7,10,12} P-={5}] />
<NP, 10, <nc-mp->, {NP8, NP9}, [P+={12} P-={6}] />
```

Edges can specify any kind of objects, from lexical to syntagmatic categories. In the former case, the argument *Nodes* only represents positions in the input and *Charac* is empty as in the examples:

```
<N, 1, <nc-fs->, {2,3}, _>
<Adj, 2, <aq-ms->, {7,8}, _>
```

The first edge specifies the lexical category tagging the word `marine`, between positions 2 and 3, and indexed by 1. The second examples describes the adjective `blanc` between the nodes 7 and 8 of the input.

This representation offers several interests. First of all, any kind of information can be represented, in a very flexible way. In particular, there is no restriction due to a strict hierarchy (a tree for example). It is then possible to represent relations between any kind of object, whatever its level. Moreover, several edges can represent different information over the same object: encoding ambiguity is then direct. But the main interest consists in the fact that such encoding

allows to extract different kind of information. It is for example possible to obtain at a very general level the set of categories (in other words the bracketing), or at a very fine level, the set of obligation relations involving for example a finite verb. Moreover, such technique allows to annotate (i.e. add new edges) at any time. It is then possible to annotate only the constituency information and to refine the granularity later by adding other edges corresponding to new properties.

3.2 Multilevel annotation

We briefly address in this section the question of multilevel annotation. This problem, in particular for annotating spoken language coprpora, is of deep importance: most of language phenomenon has to be described taking into account several domains of linguistic analysis such as phonetic, prosody and syntax. Several works (see for example [Blache00b]) has shown the difficulty of representing such information by means of strictly hierarchical structures: it is for example generally impossible to "insert" prosodic information into a syntactic tree. Constraint-based approaches constitutes an interesting and efficient answer to this problem: information in this case can be represented in a decentralized way in which each domain can form a constraint system. Insofar as constraints are not hierarchized, it is then possible to express relations between the different domains either directly between objects, using a similar representation as in the previous section, or between relations (using in this case the possibility of considering an edge label as a node).

We propose in the following example an annotation of several domains for the input:

`c'est presque impossible`

it's almost impossible

In this example, we represent phonetic, phonologic and syntactic information. We need for this to add an argument to the edge structure described in the previous section. This argument indicates the type of the domain under consideration. The XML representation of an edge is then an element of the form:

<Type, Cat, Index, Feat, Nodes, Charac />

in which the argument `Type` represents the type of the information: `PHONT, PHONL, PROS, SYNT` which respectively stand for phonetic, phonology, prosody and syntax.

```
<graph>
<sentence value= "c'est presque impossible"/>
<node id="n1" time="0"/>
<node id="n2" time="118"/>
...
<node id="n18" time="1726"/>
<PHONT, phnm, 1, /s/, <n1, n2>, _/>
<PHONT, phnm, 2, /e/, <n2, n3>, _/>
<PHONT, phnm, 3, /p/, <n3, n4>, _/>
<PHONT, phnm, 4, /R/, <n4, n5>, _/>
<PHONT, phnm, 5, /E/, <n5, n6>, _/>
<PHONT, phnm, 6, /s/, <n6, n7>, _/>
<PHONT, phnm, 7, /k/, <n7, n8>, _/>
<PHONT, phnm, 8, /e/, <n8, n9>, _/>

...
```

```
<PHONL, syl, 18, /se/, <n1, n4>, _/>
<PHONL, syl, 19, /pRE/, <n4, n6>, _/>
<PHONL, syl, 20, /skẽ/, <n6, n10>, _/>
...
<PROS, TU, 26, _, <n4, n12>, _/>
<PROS, TU, 27, _, <n12, n18>, _/>
...
<PROS, tone, 29, M, <n1, n4>, _/>
<PROS, tone, 30, T, <n4, n12>, _/>
...
<SYNT, Pro, 33, ``c'', <n1, n3>, _/>
<SYNT, V, 34, ``est'', <n3, n4>, _/>
<SYNT, Pro, 35, ``presque'', <n4, n10>, _/>
...
<SYNT, NP, 37, _, <n1, n2>, _/>
<SYNT, VP, 38, _, <n3, n18>, _/>

</graph>
```

We only present in this example the main structure, without describing the respective characterizations. Moreover, the argument `Feat` can have different semantics according to the domain. In some cases, it represents the realization of the object (for example the phoneme), in other cases, it bears a description of the internal characteristics of the object (morpho-syntactic features for example).

The first part of the annotation contains the specification of the nodes. As indicated before, a node (at least at the first stage of the annotation) is simply a position in the acoustic signal. It associates an index with an absolute position (here in milliseconds). The different edges connect these nodes.

This example shows how different domains can be represented by means of graphs in XML. It illustrates the interests of graphs, in particular the possibility of representing structured information without the constraint of strict hierarchization. Moreover, one can easily imagine annotating separately the different levels, which is imperative, the annotating tools (and the experts) being differents. This multilevel annotation is an ongoing project corrdinated by the LPL, in collaboration with several laboratories working on corpus description, the development of annotating tools or in the field of database theory (the PRAX project, see `http://www.lpl.univ-aix.fr/PRAX`).

4 Syntactic annotations of spoken language corpora

There are several questions related to syntactic annotation of spoken language corpora, among which the choice of the mode of transcription and the representation and treatment of the paradigmatic structures. We briefly adress in this section these questions before presenting an example.

4.1 Problems of transcription

Before annotating a spoken language corpus, it is obviously necessary to transcribe it. The most frequent transcription method in the perspective of syntactic annotation is the spelling writing. However the spelling, while being more legible than a phonetic transcription, does not allow a faithful reproduction of syntactic elements:

- the spelling does not provide acoustic information. For example, transcribing the word `plus` (*more*) without phonetics doesn't allow to know whether it was pronounced `[ply]`, `[plyz]` or `[plys]` while it can imply different categorizations, with different properties: it is an adverb when pronounced [plyz], but a comparative degree when pronounced `[plys]`;

- on the other hand, it introduces elements which are not produced in speaking (as most of the plural “s” for nouns: avion (*plane*) [avjõ] vs. avions (*planes*) [avjõ]
- finally, it can represents in a different way morphological divisions (in the opposition between grand and grande (*tall*) the grapheme “e” which marks the feminine, while in the oral the opposition is made on the pronunciation or not of [d]: [gRã] vs. [gRãd])

It seems then reasonable to adopt a double transcription, containing a spelling part for more comfort of reading, and a phonetic part for more exactness in the transcription. Furthermore in the perspective of enriching afterward the annotations by information of different levels, it is important to have both types of transcription at our disposal.

4.2 Representation and treatment of paradigmatic structures

This information type is of deep importance for annotating spoken language corpora. It reflects a type of organization which overlaps in the traditional phrasal progress. The phenomenon can take on several forms: fragments, hesitations, false starts, repairs, etc., typical of the discourse level.

Example:

dès l’arrivée sur cette frontière qui est blafarde qui est sinistre véritablement sinistre comme toutes les frontières

from the arrival at the border which is wan which is sinister absolutely sinister as all borders

One can represent, according to recommendations presented in [Blanche-Benveniste97], this type of information in a vertical way, highlighting the paradigmatic progress of the syntactic structure:

```
dès l’arrivée sur cette frontière qui est              blafarde
                                    qui est            sinistre
                                            véritablement sinistre
                                                          comme toutes les frontières
```

4.3 Example in XML

Syntactic information should be noted on the double transcription (spelling and phonetic) of the input, cut in coherent minimal categories for this kind of analysis. The following figure proposes an example of syntactic annotation indicating some of edges and their representation:

bon bon alors voilà voilà donc il s’agit de d’une expérience que nous ont commandité les sociologues hein

well well then that’s it thats’ it so this is a an experience that sociologists order us OK

```
<graph><sentence>
<level val="0">
  <arc cat="adj" index="1" feat="afpms-" nodes=1,1> bon </arc>
  <arc cat="adv" index="2" feat="rgp" nodes=1,1> bon </arc>
  <arc cat="adv" index="3" feat="rgp" nodes=2,2> alors </arc>
  <arc cat="prep" index="4" feat="sp-" nodes=3,3> voilà </arc>
  <arc cat="prep" index="5" feat="sp-" nodes=4,4> voilà </arc>
  <arc cat="conj" index="6" feat="cc-" nodes=5,5> donc </arc>
  <arc cat="pro" index="7" feat="ppnmst-" nodes=6,6> il </arc>
  <arc cat="N" index="8" feat="nc-mp-" nodes=7,7> s’agit </arc>
```

```
  <arc cat="V" index="9" feat="vmi-stp" nodes=8,8> de </arc>
  <arc cat="prep" index="10" feat="spx" nodes=9,9> d'une </arc>
  <arc cat="N" index="11" feat="nc-mp-" nodes=10,10> expérience
</arc>
...
</level><level val="1">
  <arc cat="Sadv" index="26" feat="rgp" nodes=1,1> bon </arc>
  <arc cat="SA" index="27" feat="afpms-" nodes=1,1> bon </arc>
  <arc cat="Sadv" index="28" feat="rgp" nodes=2,2> alors </arc>
  <arc cat="SN" index="29" feat="ppnmst-" nodes=6,6> il </arc>
  <arc cat="SV" index="30" feat="vmi-stp" nodes=8,8> de </arc>
  <arc cat="SP" index="31" feat="spx" nodes=9,10> d'une expérience
</arc>
  <arc cat="SN" index="32" feat="pi-fs--" nodes=11,11> que </arc>
...
</level><level val="2">
  <arc cat="SN" index="41" feat="pi-fs--" nodes=9,11> d'une </arc>
...
  <arc cat="Rel" index="47" feat="pr-----" nodes=13,16> ont commandité les so-
ciologues </arc>
  <arc cat="SV" index="48" feat="vmpmpss" nodes=15,18> les sociologues hein </arc>
  <arc cat="SV" index="49" feat="vmpmpss" nodes=14,18> commandité les socio-
logues hein </arc>
</level><level val="3">
  <arc cat="SN" index="50" feat="nc-fs-" nodes=11,16> que nous ont commandité
les sociologues </arc>
  <arc cat="SN" index="51" feat="nc-fs-" nodes=11,15> que nous ont commandité
les </arc>
  <arc cat="Rel" index="52" feat="pr-----" nodes=13,18> ont commandité les so-
ciologues hein </arc>
...
</level><level val="5">
<arc cat="SV" index="63" feat="vmi-stp" nodes=8,18> de d'une expérience que nous
ont commandité les sociologues hein </arc>
</level>

</graph></sentence>
```

5 Conclusion

The question of flexibility is central for syntactic annotation. It allows on the one hand to annotate a corpus at different granularity levels. It is for example possible to annotate the bracketing information plus finer information only for *VP*. It is also possible to refine the annotation of an already annotated corpus simply in adding new edges. On the other hand, insofar as all syntactic description is represented by means of distinct properties, one can extract any information, at any level from such resources. In the same perspective, a constraint-based representation allows flexibility thanks to incrementality. It is possible to annotate different kind of information (including different granularity levels) during separate phases. In other words, it is allways possible to enrich an already annotated corpus.

At a more general level, this annotation framework allows to encode any kind of information. It is then possible to enrich such corpora with information coming from other domains such as prosody, semantics, etc.

References

[Balfourier02] Balfourier J.-M., P. Blache & T. van Rullen (2002) "From Shallow to Deep Parsing Using Constraint Satisfaction", in proceedings of *COLING-2002*.

[Bird99] Bird S. & M. Liberman (1999) "A formal framework for linguistic annotation", *Technical Report MS-CIS-99-01*. Dept of Computer and Information Science, University of Pennsylvania.

[Blache00a] Blache P. (2000) "Constraints, Linguistic Theories and Natural Language Processing", in *Natural Language Processing*, D. Christodoulakis (ed.), Lecture Notes in Artificial Intelligence, Springer.

[Blache00b] Blache P. & D. Hirst (2000) “Multi-level annotation for spoken-language corpora”, in proceedings of *ICSLP-2000*.

[Blache01] Blache P. & J.-M. Balfourier (2001) “Property Grammars: a Flexible Constraint-Based Approach to Parsing”, in proceedings of *IWPT-2001*.

[Blanche-Benveniste97] Blanche-Benveniste C. (1997) *Approches de la langue parlée en français*, Ophrys.

[Cassidy02] Cassidy S. & S. Bird (2002) “Querying Databases of Annotated Speech”, in proceedings of *11th Australian DB Conference*.

[Hirst98] Hirst D. (1998), “Intonation in British English”, in Hirst D. & A. Di Cristo (eds) *Intonation Systems*, Cambridge Universtity Press.

[Kallmeyer00] Kallmeyer L. (2000) “A query tool for syntactically annoted corpora”, in proceedings of *ACL 2000*.

[Milde02] Milde J.-T. & U. Gut (2002) “The TASX-environment: an XML-based toolset for time aligned speech corpora”, in proceedings of *LREC*.

[O’Donnell00] O’Donnell M. (2000) “RSTTool 2.4. A Markup Tool for Rhetorical Structure Theory”, in proceedings of *INLG’2000*.